UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/090,489	03/04/2002	Ranjit S. Oberoi	5681-14000	5065

7590 09/06/2006

Jeffrey C. Hood
MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL
P.O. Box 398
Austin, TX 78767

EXAMINER	
WOODS, ERIC V	
ART UNIT	PAPER NUMBER
2628	

DATE MAILED: 09/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/090,489	**Applicant(s)** OBEROI ET AL.
	Examiner Eric Woods	**Art Unit** 2628

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 June 2006*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *7-10,17-22 and 25-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *7-10,17-22 and 25-33* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Response to Arguments

Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

The Office is withdrawing the finality only because the rejections of certain dependent claims were incorrectly rejected, in that an improper dependency tree was used to structure said rejections.

Applicant's arguments with respect to all claims have been considered but are not found to be persuasive.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 7-9 are rejected under 35 U.S.C. 101 because they do not recite statutory subject matter. That is, claim 7 does not provide a concrete, tangible, and practical application. The output is not displayed to the user or stored for later display, so there is no physical transformation, or concrete, tangible, and practical application. The claim – as written – is merely manipulating information within a computer.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent

> **granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.**

Claims 7 and 17 are rejected under 35 U.S.C. 102(e) as being anticipate by Hamburg (US 6,028,583 A).

As to claim 7,

A method comprising:

(a) Reading a first stream of image pixels corresponding to an image X_K from an image memory; (Hamburg, Figure 9, layers C to (C+k) constitute images and image pixels. Notably, the image files of the Hamburg system are shown in Figure 7, where each layer contains its own image data 54 as part of layer 52. These clearly constitute an image X_K. See 4:7-22, where clearly this data is stored in some kind of memory, where that would inherently be image data since an image is stored in it. Also, the various intermediate layers constitute images on)

(b) Reading a second stream of pixels corresponding to an image A_K from an accumulation buffer; (Hamburg Figures 9 and 10 specifically, where Figure 10 initializes a set of accumulation buffers to an empty or blank state (step 82) – 4:50-65. Next, clearly an image is put into the primary accumulation buffer (see Figure 10, where if the layer is compound, step 92 "copies the primary buffer into the secondary buffer", or the other alternative, step 86 executes the step "Composite layer with primary buffer", where it is clear that buffer already contains image information of some kind, even if it is only single color information from the initialization step (4:50-65), where the compositing steps are explained in 5:1-6:55. Therefore, clearly the first intermediate image is stored in an accumulation buffer. Hamburg Figures 9 and 10, 4:34-9:10, specifically 4:34-50, "The first intermediate image is stored, e.g. in a volatile or non-volatile memory, to provide a stored intermediate image 72." See also Figure 9, first intermediate layer, consisting of layers 1 to (C-1), where this is stored in the accumulation buffer (prima facie the accumulation buffer is a type of image memory), as discussed in 4:45-5:55, as are all other intermediate images.)

(c) Blending each image pixel of the image X_K with the corresponding pixel of the image A_K based on an alpha value provided with the image pixel, and thus, generating a third stream of output pixels defining an image A_{K+1}; and (Hamburg clearly blends these, where image layers are clearly images – as shown in Figure 7, where an image 54 resides on a layer 42. These are composited, which clearly is a blending operation, where transparency treatment information 58c, image layer global opacity 58a, and the like are present to control the compositing operation for each layer. Each layer or group of layers is merged on a per-pixel basis, where images are made of pixels (1:28-40), where opacity of each layer is clearly considered (1:40-65). The system of Hamburg does so with more granularity, as shown in Figure 2, where transparency information clearly constitutes an alpha value associated with or provided with the image pixel on a particular layer. Clearly the output of this combination is an image – note in Figure 9 that second and third intermediate images are formed)

(d) Transferring the third stream of output pixels to the accumulation buffer; (Hamburg Figure 10, all the intermediate data and the like are stored in accumulation buffers, where clearly this would constitute 'transferring the output pixels to the accumulation buffer,' where all the memory / storage of the image information **is** accumulation buffers in the system of Hamburg.)

(e) Performing (a), (b), (c), and (d) for each image after the first image of a sequence of N images $X_{K,}$ for K = 0, 1, 2, ..., N-1. (Hamburg shows that the first and second intermediate images are blended to generate the third intermediate image, and that that third intermediate image is then composited with layers (C+k+1), where each layer is an image, and clearly there is a sequence of images).

Hamburg teaches all of the above limitations.

As to claim 17, Hamburg clearly teaches an image memory that consists of a plurality of accumulation buffers and otherwise volatile or nonvolatile storage as described above in the rejection to claim 7, where memory is memory 124 in Figure 12. Further, the mixing unit comprises the process 126 executing the compositing program 124, and the accumulation buffers are areas of memory 124 (4:15-50).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.

2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 7 and 17 are rejected under 35 USC 103(a) as unpatentable over Grzeszczuk et al (US 6,667,957 B2).

As to claim 7,

A method comprising:

(a) Reading a first stream of image pixels corresponding to an image X_K from an image memory; (Grzeszczuk – second texture (texture is inherently an image) from a memory (any memory containing an image or texture is inherently an image memory), where an image inherently consists of pixels and images are sent through the pipeline in stream fashion; they are not processed entirely in parallel – the system of Grzeszczuk processes them in stream fashion – 19:5-30.)

(b) Reading a second stream of pixels corresponding to an image A_K from an accumulation buffer; (Grzeszczuk clearly reads the first texture, which has been applied to the surface map and the results stored to the accumulation buffer, from the accumulation buffer. 19:5-30)

(c) Blending each image pixel of the image X_K with the corresponding pixel of the image A_K based on an alpha value provided with the image pixel, and thus, generating a third stream of output pixels defining an image A_{K+1}; and (Grzeszczuk blends / combines the results of the first texture, which was texture mapped to the surface map and stored to the accumulation buffer, with the second texture – the result of applying texture mapping to the view map, and then combining the results via pixel-by-pixel multiplication (thusly showing the 1:1 correspondence). Next, the results of such a combination are clearly output – 19:5-30, where the combining takes place with respect to alpha values – see 19:40-60 and expressly in 19:5-30, where alpha blending does take place, after the texture blend occurs which will include alpha blending *because the textures are RGBA textures, with an alpha channel; therefore, when they are combined the alpha value will be multiplied and **alpha blending** will take place since the composite texture will consist of elements from each component texture – 19:40-60, where each texture has an alpha channel.)

(d) Transferring the third stream of output pixels to the accumulation buffer; (Grzeszczuk 19:5-30, the results of the above operation are written back to the accumulation buffer, where this would constitute 'transferring the third stream of output pixels to the accumulation buffer.)

(e) Performing (a), (b), (c), and (d) for each image after the first image of a sequence of N image X_K, for K = 0, 1, 2, ..., N-1. (Grzeszczuk clearly teaches that the rendering is performed with respect to time, see 3:20-40 where variable lighting conditions, e.g. lighting that varies with time, is applied, which clearly shows that the rendering acts in stream fashion such that the stream is kept going, where clearly this would constitute a series of images.)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use an accumulation buffer for the texture buffering, since the system of Grzeszcuk can be multi-texturing based, which therefore requires that the resultant textures be accumulated when the are being composited.

As to claim 17, this is exactly the system used to execute the above. Clearly, Grzeszcuk has an accumulation buffer – 19:5-30, and it has image memories as recited in the above rejection to claim 17, which is incorporated by reference.

Claim 25 is rejected under 35 U.S. C. 103(a) as obvious over Grzeszcuk in view of Adler et al (US 6,028,907).

As to claim 25, this is essentially the same system as that of claim 17 with additional limitations, the rejection to which is incorporated by reference. Specifically, the mixing unit of the system of claim 17 is comparable to the accumulation unit of claim 25. The limitation of processing N of the images is taught in the third clause. The "2D slice" of the instant claim is comparable to the X_{Kth} image – that is, an image is inherently 2D, and so such a slice would in fact meet that limitation. Also, the weighted value is not specified to be an alpha value, so in theory Grzeszcuk alone would be sufficient to make the rejection, but the other references are included for the reasons discussed in the Response to Arguments section and the rejection to claim 17 itself.

Specifically, Adler teaches in Figure 2 that a stack of 2D slices from a CT scan (which is known in the art to be generated by incrementally moving a patient through a fixed scanning apparatus to generate a stack of sequential two-dimensional images of a 3D object) can be merged to generate a three-dimensional model of said object. Adler further marks contours on each object (as is apparent in Figure 2)(4:35-55, for example) so that a composite view of the three-dimensional object can be generated and navigated around in three-dimensional space (6:10-16).

Obviously, the system of Grzeszcuk could be used to generate the resultant three-dimensional view, since Adler does not specify that much of the specific graphics subsystem used to calculate such details.

It would be obvious to one of ordinary skill in the art at the time the invention was made to combine the systems of Grzeszcuk with the system of Adler so that a system that could more rapidly render three-dimensional models of bone deformation for scoliosis and the like could be generated and efficiently navigated through by a user.

Claims 7, 10, and 17 are rejected under 35 USC 103(a) as unpatentable over Morein in view of Haeberli and MacInnis.

As to claim 7,

A method comprising:

(a) Reading a first stream of image pixels corresponding to an image X_K from an image memory; (Morein teaches a first stream of pixels stored in drawing buffer 140 in Figure 1 – 5:4-50)

(b) Reading a second stream of pixels corresponding to an image A_K from an accumulation buffer; (Morein teaches a second stream of pixels stored in first accumulation buffer 170 in Figure 1- 5:4-50)

(c) Blending each image pixel of the image X_K with the corresponding pixel of the image A_K based on an alpha value provided with the image pixel, and thus, generating a third stream of output pixels defining an image A_{K+1}; and (Morein teaches that the controller 160 blends the contents of accumulation buffer 170 and drawing buffer 140, where Morein clearly teaches that while the combination is subject to the mask buffer 150, **the mask buffer can clearly be set so that all pixels are valid**. As such, the pixels would be combined on a one-to-one basis in that scenario.)(Haeberli clearly teaches the advantages of using RGBA channel images and accumulation buffers) MacInnis clearly provides a system that performs alpha blending 1:55-2:15, where such alpha blending can be done on a per-pixel basis, as noted above (15:47-16:5))

(d) Transferring the third stream of output pixels to the accumulation buffer; (In addition, Morein teaches that once a predetermined number of accumulation operations have occurred for a predetermined number of images, said first accumulation buffer acts as the output buffer (Co1. 6, lines 13-25). Thus, said blended pixel data is transferred back and stored in the first accumulation buffer. Then the second accumulation buffer takes over the next predetermined set of accumulation processes. In other words, the first and second accumulation buffer takes turn accumulating pixel data for each sequence of image.)

(e) Performing (a), (b), (c), and (d) for each image after the first image of a sequence of N image X_K, for K = 0, 1, 2, ..., N-1. (Morein teaches a method and apparatus for supporting accumulation buffering in a video graphics system (Col. 2, lines 3-36), which explicitly is accumulating each image in a sequence of images X_K, K=0, 1, 2, ... N-1. All video comprises a sequence of images.)

It is well known in the art that frame buffers typically store data in an RGBA (red, green, blue, alpha format). An analogous art, Haeberli et al. teaches that the accumulation buffer provides 16 bits to store each red, green, blue, and alpha color components (Pages 31 1, Section 3.2). Thus, by definition, blending the alpha color components of all images being blended specifically is blending based on the alpha value provided with each pixel. Haeberli is **ONLY** utilized to show that the accumulation buffer has RGB and alpha (A) channels. Note the discussion concerning the limitations of Haeberli in the Response to Arguments section in the previous Office Action, which is incorporated by reference. Further, Haeberli specifically sets forth on pages 311-313 the benefits of using accumulation buffering to eliminate anti-aliasing and how the implementation used by the authors (e.g. Haeberli) has certain advantages, which further provides motivation.

Alpha blending is obvious and well known in the art, as stated in the last Office Action. MacInnis is added as proof of this statement.

It would have been obvious to one of ordinary skill in the art at the time of the invention to take the teachings of Morein and to modify it by adding the method of blending each image pixel based on the alpha value of each image pixel in order to implement standard blending using the accumulation buffer. The alpha values dictate the opacity of each image pixel, which influence the weighed average of the color values for each pixel, and thus blending by alpha value is a necessary standard technique well known in the art as shown in MacInnis, since this modification would allow the creation of multiple region graphics and provide many other useful capabilities to the system of Morein, which has accumulation buffers that prima facie possess alpha values on a per-pixel basis as specified in Haeberli (and accumulation buffers have been known to have alpha values on a per-pixel basis since Haeberli, well over twelve years before the filing of the instant application, and a decade before the filing of the Morein and MacInnis application), since this allows the specification of quadruplet RGBA values for writing to the accumulation buffer, which is known to be more efficient, and for the reasons specified in the previous Office Action.

It is also respectfully noted that Morein in 2:1-45 clearly states that multiple images of a plurality or set of images can be composited together as recited by applicant in step (e), so that argument is also inapposite. Further, Morein and MacInnis both process video data, where clearly a set of images is processed as required in step (e), so under either interpretation the references of record still teach those limitations.

Specifically, the system of Morein performs blending which is known to be alpha blending (2:1-47) with respect to color, depth, transparency, and the like. Morein does not expressly teach blending on a per-pixel basis. Such a limitation is at least fairly suggested by Haeberli as above. However, since applicant insists on such being made explicit, examiner will make the MacInnis reference of record as proof that blending on a per-pixel basis is well known and obvious. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Morein in light of Haeberli and MacInnis to utilize per-pixel alpha blending since MacInnis teaches that (6:30-45, 6:59-7:15, 9:15-25, 13:25-50) such techniques produce better anti-aliasing, provide more efficient execution, use less memory, and otherwise reduce memory bandwidth, which agrees with Haeberli, which also teaches a per-pixel weight factor (even though applicant continues to refuse to acknowledge this point).

In reference to claim 9, Morein, Haeberli, and MacInnis teach the method of claim 7 as described above. In addition, Morein and Haeberli et al. teach in combination said blending comprises blending red, green and blue components of each output pixel in parallel.

As applied to claim 7 above, all color values are blended together for each pixel (Col. 5, line 26 - Col. 6, line 25). Since Haeberli et al. teaches that accumulation buffers provides storage for RGB and alpha color values, blending color values together as taught by Morein specifically is blending RGB and alpha values together. And since all color values for each pixels are blended together, said blending of RGB and alpha values must be performed in parallel.

In reference to claim 10, Morein, Haeberli, and MacInnis teach the method of claim 7, but do not explicitly teach wherein (a), (b), (c), (d) and (e) are performed by a graphics hardware accelerator chip in response to software functions executed on a host processor. Although Morein and Haeberli et al. do not explicitly name a hardware accelerator, the prior art discloses the overall architecture of one embodiment where the system is implemented on a processor, a state machine, or other circuitry (Col. 5, lines 26-30 and Col. 7, lines 6-16). Said graphics system comprising an integrated circuit specifically is a hardware accelerator chip performing in response some sort of software commands to perform (a), (b), (c), (d), and (e). Also, the instruction is given via software since that is the conventional method of communication between the processor and the rest of the graphics hardware. Thus, it would have been obvious to implement the graphics system of Morein and Haeberli et al. using a hardware accelerator chip. Further, take note of MacInnis, which clearly shows a hardware accelerator chip as part of the set-top box as noted above, see Figures 1-5 as an illustration of this. MacInnis is used to show such a hardware accelerator chip, and also (as in the rejection to claim 7 above) as evidence that alpha blending is well known in the art and has certain benefits.

In regards to claim 17, the same basis and rationale for claim rejection as applied to claim 7 above. The limitations of claim 17 are identical to the limitations of claim 7, except for one added limitation directed to the mixing unit. Morein explicitly teaches a mixing unit (Col. 5, line 26 - Col. 6, line 25). Said controller (160) specifically is a mixing unit. MacInnis also teaches a mixing unit, otherwise known as the blending unit, see for example (note Fig. 28, specifically Figure 3, the video compositor unit 60 contains the blending unit).

Claims 8, 18-19, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morein in view of Haeberli and MacInnis, and further in view of McReynolds.

In reference to claims 8 and 18, Morein, Haeberli, and MacInnis teach the method of claim 7, but does not explicitly teach the color precision of the accumulation buffer is greater than the color precision of the image buffer. It is well known and obvious, however, to implement a more precise output data calculation in order to avoid losing original data precision and minimize aliasing. An analogous art, McReynolds et al., teaches said limitations.

- McReynolds et al. teaches that 'in order to maintain accuracy over many blending operations, the accumulation buffer has a higher number of bits per color components than a typical color buffer (section 6.4, lines 3-4). Higher number of bits per color components will result in greater color precision for the accumulation buffer.

It would have been obvious to someone of ordinary skill in the art to take the teachings of Morein, Haeberli, and MacInnis and to add from McReynolds, the method of providing higher color precision of the accumulation buffer than the color precision of the image buffer in order to maintain color precision accuracy over many blending operations. This prevents loss of data and alleviates aliasing problems. It is always important to maintain precise accuracy of data after any data processing.

In reference to claim 19, Morein, Haeberli, and MacInnis teach the method and system of claim claims 7 and 17 above and Morein, Haeberli, and MacInnis and McReynolds teach the system of claims 8 and 18. In addition, remember that Morein teaches a first and a second accumulator in order to minimize the delay between accumulation and rendering. Since each pixel provides a RGB color component and an alpha value, each of the plurality of accumulators is capable of mixing a corresponding color component. Morein also explicitly teaches that if the color data includes multiple color portions, such as red, green, and blue portions, each of these portions will be treated individually by the output block (Col. 6, lines 37-42), and thus it would have been obvious to one of ordinary skill in the art at the time of the invention to take the teachings of Morein and Haeberli and to implement a plurality of mixing units to accumulate individual color components. Since parallel processing is well known and obvious in the ad, it would have been obvious to use a plurality of mixing units to comprise the controller (160) of Morein. Haeberli et al further suggests this, since an accumulator buffer comprises 16 bit to store each red, green, blue, and alpha components, it would be wise to apply a different mixer for each component in order to perform parallel processing and speed up the overall image processing.

In reference to claim 22, Morein, Haeberli, and MacInnis teach the system of claim 17, and Morein, Haeberli, MacInnis, and McReynolds teach the system of claim 18 above. While Morein, Haeberli, and MacInnis do not explicitly teach the color precision of the accumulation buffer is at least ΔN larger than the color precision of the image buffer, wherein ΔN is the base two logarithm of the maximum number of images to be blended into the accumulation buffer, McReynolds et al. teaches said limitation in the following in similar fashion as applied to claims 8 and 18 above.

Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morein in view of Haeberli and MacInnis as applied to claim 17, and further in view of Murata and Takeuchi.

In reference to claim 20, Morein, Haeberli, and MacInnis teach the system of claim 18 above, but do not explicitly teach the accumulation buffer resides within a texture buffer of a graphics system. But, remember that Murata explicitly teaches that a plurality of buffers can reside in one large buffer unit (see Figure 4 as an example, where instead of the plurality of RAMs as in Figure 3, one can be used, as explained in 3:15-45). An analogous art, Takeuchi, explicitly teaches one memory module comprising a plurality of buffers including an accumulation buffer (47) and a texture buffer (48) connected as one unit (FIG. 3). Since the accumulation buffer and the texture buffer are indeed connecting together in the figure, Takeuchi explicitly teaches an accumulation buffer residing within a texture buffer. In addition, in light of the well know and standard memory allocation technique as taught by Murata as ideal (Figure 4, Figure 12, 9:37-65), it would have been obvious to one of ordinary skill in the ad at the time of the invention to take the teachings of Morein, Haeberli, and MacInnis, and to add from Murata, and Takeuchi the memory allocation technique to combine a plurality of buffers in one large memory module in order to save space and speed up data transfer as applied to claim 21 above. This effectively eliminate the need for extra individual buffers and to expand the capacities of the texture buffer since a texture buffer can include several SDRAMS capable of housing several types of buffers and memories. Further, having the accumulation buffer reside in the texture buffer will reduce interconnect lengths and thus improve speed and efficiency of the hardware accelerator.

Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morein in view of Haeberli and MacInnis, and further in view of Murata et al.

In regards to claim 21, Morein, Haeberli, MacInnis teach the system of claim 17 above, but do not explicitly teach wherein the image buffer resides within the frame buffer of a graphic system. It is, however, well known in the art that frame buffer is a memory module storing image information to be sent to the display device (e.g. Monitor), and it is also well known and standard in the art the a memory module can comprise a plurality of separate memory units. This allows for easy transfer of data from one memory to another, especially any image data from image buffer to frame buffer for the purpose of speedy display of said image data. For example, an analogous art, Murata et al. explicitly teaches that a frame buffer comprises an image buffer and a Z buffer (Col. 1, lines 33-40, Col. 3, lines 4-37 and FIG. 1(A), 3-4). It would have been obvious to one of ordinary skill in the art at the time of the invention to take the teachings of Morein, Haeberli, and MacInnis, and to add from Murata et al., the combined image and frame buffer since it is well known and obvious standard in the art. Having separate buffers in one memory (buffer) module saves space, speeds data transfer and provides overall efficient graphic system.

Claims 25-26 and 28-29 are rejected under 35 U.S. C. 103(a) as obvious over Morein, Haeberli, and MacInnis as applied to claim 17, and further in view of Adler et al (US 6,028,907).

As to claim 25, this is essentially the same system as that of claim 17 with additional limitations, the rejection to which is incorporated by reference. Specifically, the mixing unit of the system of claim 17 is comparable to the accumulation unit of claim

25. The limitation of processing N of the images is taught in the third clause. The "2D slice" of the instant claim is comparable to the X_{Kth} image – that is, an image is inherently 2D, and so such a slice would in fact meet that limitation. Also, the weighted value is not specified to be an alpha value, so in theory Morein alone would be sufficient to make the rejection, but the other references are included for the reasons discussed in the Response to Arguments section and the rejection to claim 17 itself.

Specifically, Adler teaches in Figure 2 that a stack of 2D slices from a CT scan (which is known in the art to be generated by incrementally moving a patient through a fixed scanning apparatus to generate a stack of sequential two-dimensional images of a 3D object) can be merged to generate a three-dimensional model of said object. Adler further marks contours on each object (as is apparent in Figure 2)(4:35-55, for example) so that a composite view of the three-dimensional object can be generated and navigated around in three-dimensional space (6:10-16).

Obviously, the system of Morein could be used to generate the resultant three-dimensional view, since Adler does not specify that much of the specific graphics subsystem used to calculate such details.

It would be obvious to one of ordinary skill in the art at the time the invention was made to combine the systems of Morein, Haeberli, and MacInnis with the system of Adler so that a system that could more rapidly render three-dimensional models of bone deformation for scoliosis and the like could be generated and efficiently navigated through by a user.

As to claim 26, Haeberli and MacInnis clearly teach the use of alpha values, and that each pixel has its own alpha value. It would be obvious that if images were being alpha-blended, that each pixel would have its own alpha value, and alpha is inherently a transparency value.

As to claims 28 and 29, since as applicant has pointed out in the Remarks on page 2, alpha is always a positive value, less than or equal to one (inclusive of zero), MacInnis inherently teaches this limitation.

Claim 29 is a duplicate of claim 28 and rejected accordingly.

Claims 27 and 30-33 are rejected under 35 U.S. C. 103(a) as obvious over Morein, Haeberli, MacInnis, and Adler as applied to claim 25 above, and further in view of McReynolds.

As to claim 27, applicant is trying to claim equation that fundamentally underlies alpha blending. The following blend operation takes place for each color channel. That equation is as follows (see McReynolds page 112, section 10.2, alpha blending, as one of an infinite number of examples of this equation):

$$C_{out} = C_{src} * A_{src} + (1 - A_{src}) * C_{dst}$$

Where C_{out} is the output color to the framer buffer, A_{src} is the alpha value, C_{dst} is the destination color, and C_{src} is the source color, where source color is the color of the overall scene and the destination color is the color of the object to be added or composited with the overall scene or present image. The equation is well known in the art.

The following equivalencies exist between the variables of the equation of applicant and the variables stated in the alpha blending equation from McReynolds: A_{K+1} is equivalent to C_{out}, alpha is equivalent to A_{src}, X_K is equivalent to C_{src}, and A_K is equivalent to C_{dst}. ($A_{K+1}=C_{out}$, alpha=A_{src}, $X_K=C_{src}$, $A_K=C_{dst}$). Now, applicant's equation will be factored and rearranged as below:

$$A_{K+1} = alpha * (X_K - A_K) + A_K \Rightarrow \quad A_{K+1} = alpha * X_K + (-alpha + 1) * A_K \Rightarrow$$

$$A_{K+1} = alpha * X_K + (1 - alpha) * A_K$$

Compare to alpha blending equation as above: $C_{out} = C_{src} * A_{src} + (1 - A_{src}) * C_{dst}$

They are exactly the same once the mappings specified above are performed. Therefore, since the MacInnis reference teaches alpha blending, it inherently teaches this limitation.

As to claims 30 and 32,

Mcrein, Haeberli, and MacInnis do not teach that the 2D images are slices of a three-dimensional object.

Specifically, Adler teaches in Figure 2 that a stack of 2D slices from a CT scan (which is known in the art to be generated by incrementally moving a patient through a fixed scanning apparatus to generate a stack of sequential two-dimensional images of a 3D object) can be merged to generate a three-dimensional model of said object. Adler further marks contours on each object (as is apparent in Figure 2)(4:35-55, for example) so that a composite view of the three-dimensional object can be generated and navigated around in three-dimensional space (6:10-16).

Obviously, the system of Morein could be used to generate the resultant three-dimensional view, since Adler does not specify that much of the specific graphics subsystem used to calculate such details.

It would be obvious to one of ordinary skill in the art at the time the invention was made to combine the systems of Morein, Haeberli, and MacInnis with the system of Adler so that a system that could more rapidly render three-dimensional models of bone deformation for scoliosis and the like could be generated and efficiently navigated through by a user.

As to claims 31 and 33, applicant is trying to claim equation that fundamentally underlies alpha blending. The following blend operation takes place for each color channel. That equation is as follows (see McReynolds page 112, section 10.2, alpha blending, as one of an infinite number of examples of this equation):

$$C_{out} = C_{src} * A_{src} + (1 - A_{src}) * C_{dst}$$

Where C_{out} is the output color to the framer buffer, A_{src} is the alpha value, C_{dst} is the destination color, and C_{src} is the source color, where source color is the color of the overall scene and the destination color is the color of the object to be added or composited with the overall scene or present image.

The following equivalencies exist between the variables of the equation of applicant and the variables stated in the alpha blending equation from McReynolds: A_{K+1} is equivalent to C_{out}, alpha is equivalent to A_{src}, X_K is equivalent to C_{src}, and A_K is equivalent to C_{dst}. ($A_{K+1}=C_{out}$, alpha$=A_{src}$, $X_K=C_{src}$, $A_K=C_{dst}$). Now, applicant's equation will be factored and rearranged as below:

$A_{K+1} = alpha * (X_K - A_K) + A_K$ => $A_{K+1} = alpha * X_K + (-alpha + 1) * A_K$ =>

$A_{K+1} = alpha * X_K + (1 - alpha) * A_K$

Compare to alpha blending equation as above: $C_{out} = C_{src} * A_{src} + (1 - A_{src}) * C_{dst}$

They are exactly the same once the mappings specified above are performed. Therefore, since the MacInnis reference teaches alpha blending, it inherently teaches this limitation.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Woods whose telephone number is 571-272-7775. The examiner can normally be reached on M-F 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Eric Woods

August 10, 2006

ULKA CHAUHAN
SUPERVISORY PATENT EXAMINER